MAGPIE TUTORIAL

Configuration and usage

Abdoulaye Gamatié, Pierre-Yves Péneau

LIRMM / CNRS-UM, Montpellier

ComPAS Conference, June 2017, Sophia-Antipolis

Other contributors: S. Senni, T. Delobelle, Florent Bruguier, L. Torres, G. Sassatelli

PROJET FINANCÉ PAR L'ANR – PROJECT FUNDED BY THE ANR

Roadmap

- Setting up MAGPIE:
  - Configuring the environment
    - Output folder for results, …
  - Configuring the simulated system
    - Number of CPUs, memory hierarchy, frequency, …
- Running MAGPIE
- MAGPIE outputs

SETTING UP MAGPIE

Configuration file

- Configuration is done through one file
  - Example in `$MAGPIE/magpie/config.ini.example`
- Split into two distinct parts

  1/ Environment configuration
  - Paths to tools involved in MAGPIE (e.g. gem5)
  - Simulation output

  2/ Simulated system configuration
  - Cores, memory, application, …
- Copy this file in your `$HOME` and open it

```
$ cp $MAGPIE/magpie/config.ini.example   \
  $HOME/config.ini
```

Environment configuration

- Result directories (**to edit**)
  - `result-root=/home/etu-f_compas2017-XX/results`
  - `nvsim-result-dir= =/home/etu-f_compas2017-XX/results/nvsim`

- Paths to MAGPIE's tools (**to edit**)
  - NVSim
  - Gem5
  - McPAT
  - A parser to convert gem5's output to McPAT's input

Simulated system configuration

- Operating system
- Application
- Architecture
  - Cores
  - Memory hierarchy
  - …

Operating system configuration

- `kernel=` path to Linux binary
  - Default kernel provided by gem5's team
  - You can compile your own kernel (see MAGPIE user guide)
- `machine-type=` the platform that you want to emulate, which defines external devices (configured for ARM-32)
- `dtb-file=` path to a Device Tree Blob/Binary
  - A binary that describes the underlying platform
  - ARM + Linux specific
- `disk-image=` path to the hard drive to use

Application configuration

- Two parameters: benchmark and script
- `script=` path to the `rcS` file that will be launched after the boot phase
  - Replace by the path to your `$HOME/app/hello_magpie.rcS`
- `benchmark=` Application's name in gem5. Do not use in this tutorial. See MAGPIE user guide for more information
- **Exclusive parameters** : use just one, comment the other

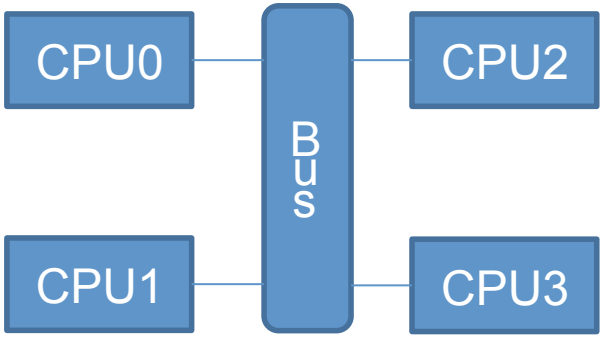
Architecture configuration: cores

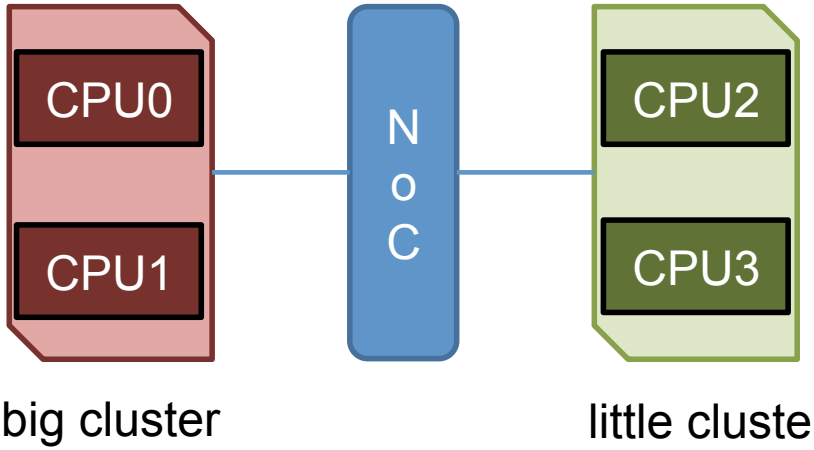
- How many CPU ?
  - `num-cpus=<number>`
- Which type of CPU ?
  - `cpu-type=<type>`
  - minor / detailed / arm_detailed / A7 / A15
  - Cortex A7/A15 are our models, calibration is based on experiments on real hardware*
- Frequency ?
  - `cpu-clock=<XX>GHz`

*** Full-System Simulation of big.LITTLE Multicore Architecture for Performance and Energy Exploration**.
Anastasiia Butko et al. IEEE 10th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC), 2016

Architecture configuration: clustering

- Two distinct modes
  - Homogeneous : cores are the same
  - Heterogeneous : two cores family in the same architecture (clustering)





- **Default: homogeneous mode**
  - Uncomment the "**big-little**" option to activate the heterogeneous mode
  - All options related to homogeneous mode will be ignored

Architecture configuration: memory

```
# *** Memory configuration *** #

# Main memory emulated in gem5
mem-type=LPDDR3_1600_x32
mem-size=2GB

# Size of one cache line/block (in bytes)
cacheline-size=64
# Technology node used for area/power estimation.
# Used only for SRAM caches, otherwise it uses the
# per cache technology configuration (in nanometers)
technology-node=22

# By default, l2 and l3 caches are deactivated. Then,
# all options related to these caches are ignored,
# even if uncommented. Uncommented the following
# option to activate l2 and/or l3 cache(s).
#l2
#l3
```

Architecture configuration: memory

L1 instruction

```
l1i-type=SRAM
l1i-size=32kB
l1i-assoc=2
l1i-technology=22
```

L1 data

```
l1d-type=SRAM
l1d-size=64kB
l1d-assoc=4
l1d-technology=45
```

L1 instruction big

```
l1i-type-big=SRAM
l1i-size-big=32kB
l1i-assoc-big=2
l1i-technology-big=22
```

L1 data big

```
l1d-type-big=SRAM
l1d-size-big=64kB
l1d-assoc-big=4
l1d-technology-big=45
```

L1 instruction little

```
l1i-type-little=SRAM
l1i-size-little=32kB
l1i-assoc-little=2
l1i-technology-little=22
```

L1 data little

```
l1d-type-little=SRAM
l1d-size-little=64kB
l1d-assoc-little=4
l1d-technology-little=45
```

Architecture configuration: memory

L2 cache

```
l2-type=SRAM
l2-size=2MB
l2-assoc=8
l2-technology=45
```

L2 cache big

```
l2-type-big=SRAM
l2-size-big=2MB
l2-assoc-big=8
l2-technology-big=22
```

L2 cache little

```
l2-type-little=SRAM
l2-size-little=512kB
l2-assoc-little=8
l2-technology-little=22
```

Architecture configuration: memory

L3 cache

```
# L3 configuration is the same regardless of
# the mode (homogeneous or heterogeneous)
l3-type=SRAM
l3-size=8MB
l3-assoc=16
l3-technology=32
```

Architecture configuration: memory

- MAGPIE allows exploration of emerging Non Volatile Memories (NVMs)

- Possible values for `lX-type=`
  - SRAM
  - STTRAM
  - RRAM
  - PCRAM

MAGPIE configuration summary

- One configuration file only
- Environment
  - Output folders
  - Paths to tools
- Simulated system
  - Operating system
  - Application
  - Architecture

RUNNING MAGPIE

Command line usage

- MAGPIE is developed in Python (version 2.7)
- The main script is **`magpie.py`** in **`$MAGPIE/magpie`**
- At least one argument is needed: configuration file
  - Could be relative or absolute path

```
$ python magpie.py --configuration-file config.ini
```

Do not launch this command for now !

Command line usage

- All options presented in **`config.ini.example`** could be overridden by the command line
  - Same name, just add double dash "--" at the beginning
- Useful if you have a fixed setup except for one parameter
  - Not necessary to create one configuration file per setup
- Example 1 : changing the cpu frequency

```
$ python magpie.py --configuration-file config.ini   \
  --cpu-freq 1GHz
```

- Example 2 : same setup for a benchmark suite

```
$ python magpie.py --configuration-file config.ini   \
  --script /path/to/your/rcS
```

Hello MAGPIE!

- Then, run MAGPIE :

```
$ cd $MAGPIE/magpie
$ python magpie.py                                    \
  --configuration-file $HOME/config.ini               \
  --checkpoint-dir $CHKPT/chkpt-1core-2GB
```

MAGPIE OUTPUTS

MAGPIE terminal output

```
[2017-06-21 14:59:18] INFO : MAGPIE started
[2017-06-21 14:59:18] INFO : Output folder :/home/peneaup/work/magpie_root/results/
1A15/2GHz/22_22/SRAM_SRAM/32kB_32kB/2_2/hello_magpie/20170621-145918
[2017-06-21 14:59:18] INFO : Check MAGPIE configuration
[2017-06-21 14:59:18] SUCCESS : Write configuration in file
[2017-06-21 14:59:18] INFO : Configure memory
[2017-06-21 14:59:18] INFO : Generate configuration file {/home/peneaup/work/
magpie_root/results/nvsim/22-SRAM-32kB-2.cfg}
[2017-06-21 14:59:31] INFO : L1 inst latency: 2/1 (read/write)
[2017-06-21 14:59:31] INFO : L1 data latency: 2/1 (read/write)
[2017-06-21 14:59:31] SUCCESS : Memory configured
[2017-06-21 14:59:31] INFO : Start gem5
[2017-06-21 14:59:31] INFO : Restore simulation with checkpoint : /home/peneaup/
work/magpie_root/gem5/chkpt-1core-2GB
[2017-06-21 14:59:31] INFO : You can access to the gem5's terminal with this
command: 'tail -F /home/peneaup/work/magpie_root/results/1A15/2GHz/22_22/SRAM_SRAM/
32kB_32kB/2_2/hello_magpie/20170621-145918/gem5/system.terminal'
```

MAGPIE terminal output

- In a second terminal, paste the command line

```
$ tail –F /home/../results/../system.terminal
Loading new script...
Hello MAGPIE!
```

MAGPIE terminal output

```
[2017-06-21 15:00:08] SUCCESS : gem5 simulation done
[2017-06-21 15:00:08] INFO : Start gem5 results converter
[2017-06-21 15:00:09] SUCCESS : gem5 results converted
[2017-06-21 15:00:09] INFO : Start McPAT
[2017-06-21 15:00:12] SUCCESS : McPAT simulation done.
[2017-06-21 15:00:12] INFO : Format results into csv
[2017-06-21 15:00:13] SUCCESS : Results ready
[2017-06-21 15:00:13] SUCCESS : Simulation finished. Output folder is /home/peneaup/
work/magpie_root/results/1A15/2GHz/22_22/SRAM_SRAM/32kB_32kB/2_2/hello_magpie/
20170621-145918
```

- Available in `/tmp/results` on your laptop
- Alternatively, type this:

```
$ rsync --progress etu-f_compas2017-XX@muse-          \
  login.hpc-lr.univ-montp2.fr:/home/etu-              \
  f_compas2017-XX/results /tmp
```

MAGPIE output files

- Located according to your configuration
  - `result-root=/home/etu-f_compas2017-XX/results/`
- MAGPIE creates under `result-root` a hierarchy of folders that represents your configuration

`1A15/2GHz/22_22/SRAM_SRAM/32kB_32kB/2_2/hello/time`

- **Number and type of cores + Frequency**
- **Technology + type of cache**
- **Cache size + associativity**
- **Application name**
- **Unique timestamp** (useful for repeated simulations)

MAGPIE output files

- 4 files for stats: 2 JSON and 2 CSV
  - `all_data.json`
    - all stats gathered by MAGPIE
  - `results.json`
    - only stats that are relevant (to us)
  - `results.csv`
    - same than `result.json` in CSV format
  - `detailed_results.csv`
    - a more complete version of `results.csv`
- Open `results.csv` and explore by yourself
- Summary of the system configuration : `configuration.txt`
- Simulation time information : `magpie_time.txt`
- A plot with power consumption repartition : `consumption-repartition.png`

CONCLUSION

Conclusion

- Summary of what you need
  - A Linux binary
  - A disk image that contains your application
  - An `rcS` file that automates the application execution
  - A checkpoint to accelerate your simulation
  - A MAGPIE configuration file that describes your architecture
    - Memory hierarchy (cache size, mem size, …)
    - Core models, organization (clustering)
    - Kernel and disk image

Conclusion

- The terminal output of the simulated system can be displayed during the execution

- Outputs are in a structured path and contains:
  - Stats results (CSV and JSON files)
  - Summary of the configuration
  - Simulation time information
  - Power consumption plot

- MAGPIE could be run in parallel as many times as you want